**Metadata on blockchain transactions:**

* **Additional data or information** that can be **attached to a cryptocurrency transaction** on a blockchain is **called metadata** in blockchain transactions.
* While the primary function of blockchain is to document and authenticate the transfer of digital assets, including cryptocurrencies, metadata allows additional information or context to be added to transactions.
* Metadata is data about data. In the context of blockchain transactions, they include information that is not directly related to the cryptocurrency transfer but **may provide additional functionality to the transaction**.
* There are **two main types of metadata** in transactions:

1. On-chain metadata 2. Off-chain metadata

**On-chain metadata**

* Since this metadata is immediately **stored on the blockchain (on-chain),** it is **part of the transaction data stored** there.
* **Anyone with access to the blockchain can see this**. Information such as **transaction labels, notes, or references to external contracts or documents can be found** in on-chain metadata.

**Off-chain metadata**

* This **data is referenced in the transaction but is not stored directly on the blockchain (off-chain).**
* Off-chain metadata may include **links to other content, such as files, documents, or web addresses** that **contain more detailed information about the transaction**. Off-chain metadata is a tool that can be **used to reduce blockchain clutter**.

**How on-chain metadata is stored:**

* **On-chain metadata**, such as transaction data, smart contract code and token properties, are an integral part of the blockchain data structure, **permanently stored and replicated in network nodes**, being a component of the distributed ledger.
* Most on-chain metadata is **stored within the transactions themselves**. In a blockchain, every transaction has a **payload with associated metadata**.
* For example, when sending cryptocurrency between addresses, the sender, recipient, amount and date of the transaction are recorded as on-chain metadata.
* When it comes to smart contracts, **the contract code and associated data** are stored on the blockchain as on-chain metadata. This includes functions, state variables, and associated contract data.
* **Each block has headers containing certain metadata that is easily verifiable, such as block timestamps, block numbers, and transaction IDs**.
* **For tokens** such as ERC-721 and ERC-20 in Ethereum, metadata about the **properties of the tokens (token name, symbol, emission) is often stored** on the blockchain.

**How Off-chain metadata is stored:**

* Off-chain metadata is **stored off-chain but cryptographically linked to it**, **increasing the efficiency and flexibility of the blockchain**.
* They can be stored in a variety of places, including off-chain scaling solutions such as the **Lightning Network, decentralized storage systems** such as the **InterPlanetary File System (IPFS), and external databases**.
* Blockchain uses **cryptographic hashes or pointers to reference off-chain metadata,** which **frees up space on the blockchain for storing more or less sensitive data, reduces congestion, and provides flexibility for applications** that need private or dynamic information.

**Examples of metadata in transactions**

Metadata on the blockchain includes **timestamps, transaction details, smart contract data, digital signatures, gas fees, IPFS references, oracle information, and non-fungible token (NFT) metadata**, allowing for various functionality and information storage. blockchain networks.

**Timestamps**

* Each block of the blockchain contains a timestamp, which **indicates when the block was added to the chain**. Using this metadata, the time of the transaction is recorded.

**Transaction Details**

* **Sender and recipient addresses, transaction amounts, and individual transaction IDs** are just a few examples of metadata that can be included in every transaction on the blockchain.

**Smart contract data**

* **The parameters and inputs required for the contract** to operate can be included in the metadata when executing smart contracts on the blockchain.

**Digital signatures**

* To confirm the **legitimacy(validity) of transactions and demonstrate ownership**, the metadata contains digital signatures.

**Gas fee**

* On blockchains such as Ethereum, metadata may contain information about **network fees associated with processing transactions**.
* Miners and validators need this information **to prioritize transactions**.

**Interplanetary File System Links**

* References to IPFS, a decentralized file storage system, can be found in the blockchain metadata. **Users can access data on the blockchain by retrieving a link (usually a hash) to the IPFS file when needed**.
* Using this method, **large files can be stored, including images, videos, or documents associated with blockchain assets such as NFTs.**

**Oracles**

* Oracles are **external services that provide smart contracts with access to real-world data**.
* **Information about these oracles can** be included in blockchain metadata to **trigger smart contract actions**.

**Non-fungible token metadata**

* NFTs often contain metadata, such as **author attribution, description, and other information about the digital or physical asset they represent**.

**How to Add Metadata to a Blockchain Transaction**

When a user wants **to add metadata to a transaction**, they can do so using a smart contract, which is a **self-executing contract with predefined conditions encoded into it**.

Using the example of Ethereum, which is known for its ability to include metadata in transactions using smart contracts, we can look at the process itself.

**Creating a smart contract**

Before adding metadata, you need to create a smart contract. Instructions about where and how the metadata will be stored are included in this contract. Metadata can be stored in a specific variable, usually as a string.

A computer screen shot of a program code

Description automatically generated

Simplified example written in Solidity language.

In the image above, the smart contract called MyContract has a public metadata variable and a setMetadata function that allows the metadata to be updated.

**Interaction with a smart contract**

* To add metadata to a transaction, the **user must run the smart contract, initializing the transaction, that is, interact**. This can be done using libraries such as **web3.js or ethers.js, or through Ethereum wallet applications**.

**Metadata verification**

By interacting with the smart contract and reading the metadata variable, anyone can confirm the metadata once the transaction is confirmed and added to the blockchain. However, it is important to remember that things to consider when adding metadata to a transaction include gas price, security, and privacy.

**Use Cases for Blockchain**

Metadata has applications in a wide range of industries, including supply chain management, digital identity, smart contracts, NFTs and healthcare.

**Supply chain management**

Businesses can improve **traceability and transparency by placing metadata** **about production, transportation, and product quality on the blockchain**. For example, a food manufacturer can track information about the origin of materials, their passage through the supply chain, and quality checks.

This information is needed to **resolve issues such as product fraud or rejection, ensure regulatory compliance, and verify authenticity**. Buyers can use this metadata to make informed decisions about the products they purchase.

**Digital identity and authentication**

Metadata can be used **to securely manage and store credentials and personal data**. People are responsible for their data and can choose to allow or deny access to it, reducing the likelihood of identity theft and privacy violations. Businesses, governments, and educational institutions can use this technology to **improve the security of services and speed up identity verification processes.**

**Self-executing contracts**

Another area where blockchain metadata plays an important role is in self-executing contracts. Metadata is used in them **to make decisions about when and how to fulfill a particular condition**. For example, a smart contract in the **insurance industry could use meteorological data as metadata to immediately trigger payments to policyholders affected by severe weather conditions.**

In the financial industry, **loan agreements can determine loan eligibility and interest rates based on credit scores and transaction history,** which are stored as metadata, allowing for automated and more efficient lending procedures.

**Non-fungible tokens and digital assets**

NFTs and digital assets often use metadata **to give digital collectibles, artwork and assets meaning and value**. **Creator information, ownership history, and characteristics of a digital object are examples of metadata**. Provenance tracking, art authentication, and the creation of decentralized applications (dApps) based on NFTs can benefit greatly from this knowledge.

**Medicine and data security**

In healthcare, metadata is **used to securely maintain patient records and ensure data integrity**.

**Patient records, patient consent forms, and data access logs can be stored in metadata, increasing the security and confidentiality of medical information**. This facilitates **communication between different healthcare systems** and allows emergency responders to obtain vital medical information faster.

**Metadata Issues**

**Scalability and storage cost issues** are important because as blockchain networks grow in size, they can become less efficient and require more resources. Large amounts of data storage on the blockchain can put stress on the network architecture and increase the risk of centralization.

Another concern is **data security and confidentiality**, **especially when working with private or highly sensitive data**. Blockchain transparency may conflict with privacy regulations, so its implementation and design must be carried out with extreme care. Additionally, the reliability of blockchain applications and smart contracts can be affected by malicious or erroneous data entry.

Solving these problems is of particular importance for the further development and implementation of blockchain technology in various industries. Overcoming these obstacles in blockchain **metadata requires a diversified strategy**. **Developers can use layer-2 solutions and sharding techniques to offload specific data from the main chain to avoid scalability issues.**

**To increase security and confidentiality, encryption and private blockchains can be used**. The reliability of an oracle can be ensured by using multiple data sources for verification and reputation mechanisms. Blockchain metadata can be made more efficient and secure through robust security protocols, innovative technologies, and careful design.